

Tallinna Ülikool
Informaatika instituut

Google AJAX search API

Seminaritöö

Autor: Rome Tikas

Juhendaja: Jaagup Kippar

Kaitsmisele lubatud:

Tallinn 2009

SISUKORD

SISUKORD.....	2
SISSEJUHATUS.....	4
1. TEOREETILINE TAUST.....	5
1.1 Google missioon.....	5
1.1 Google (SOAP) Search API.....	5
1.2 Mida tähendab AJAX	6
1.3 Google AJAX search API.....	6
1.4 AJAX search API võtme kasulikkus.....	7
1.5 AJAX search API arhitektuur.....	7
1.6 Täiustatud kasutajaliidese otsingud.....	8
1.7 Google AJAX search API Wizards.....	10
1.8 Google AJAX Search API't internetis.....	10
2. PRAKTILINE TEOSTUS.....	11
2.1 "Hello, World".....	11
2.1.1 Otsingu kood.....	11
2.1.2 Harjutusi õpilasele.....	14
2.1.3 Kokkuvõtte „Hello, World“ otsingule.....	15
2.2 Kaardi viisard.....	15
2.2.1 Probleemid kaardi viisardiga.....	15
2.2.2 Otsingu tulemus	16
2.2.3 Kokkuvõtte kaardi viisardile.....	16
2.3 Veebileht kasutades Google AJAX search API't.....	16
2.3.1 Tallinna Ülikool meedias lehekülje kood.....	17
2.3.2 Harjutusi õpilasele.....	19
KOKKUVÕTE.....	20
KASUTATUD KIRJANDUS.....	21
LISAD.....	22
Lisa 1. Lähtekood "Hello, World" jaoks.....	23

Lisa 2. Kaardi viisardiga saadud lähtekood.....	24
Lisa 3. Veebileht „Tallinna Ülikool meedias“.....	26

SISSEJUHATUS

Iga veebilehekülg on suuremal või vähemal määral sõltuv külastajate arvust. Mida suurem on külastajate arv, seda rohkem on antud leheküljel võimalus teenida oma eesmäärke. Samas on oluline, et lehekülgedel olev info ei muutu külastaja jaoks tüütuks, vastasel korral sunnib see inimest kasutama muid võimalusi leidmaks infot. Samas on oluline, et lehekülgedel asetsev info ei muutuks külastaja jaoks tüütuks, mis omakorda sunniks neid kasutama muid võimalusi info leidmiseks. Et oma külastajaid hoida, on paljud veebilehtede omanikud hakanud kasutama Google AJAX search API tooteid. Kellele on Google AJAX search API tooted kasulikud? Arvan, et on kasu kolmepoolne. Võidab Google, kui tema otsinguid kasutatakse. Võidab veebilehe omanik, kui kliendid külastavad nende lehte. Võidab ka klient, kes saab ilma veebilehelt lahkumata vajaliku info kätte. Samas kui Google AJAX search API on halvasti seadistatud, võib see kahjustada lehekülje mainet ning pärssida populaarsust.

Käesoleva seminaritöö eesmärgiks on tutvustada õppivale veebiprogrammeerijale, mis on Google AJAX search API a kuidas seda efektiivselt rakendada.

Seminaritöö koosneb kolmest osast, mis jaguneb omakorda alateemadeks. Esimeses peatükis on ülevaade Google AJAX search API kohta. Teises peatükis antakse ülevaade erinevatest praktilistest teostustest. Seminaritöö kolmas peatükk võtab tulemused kokku ning esitab mõningad ettepanekud veebilehtede programmeerijatele.

1. TEOREETILINE TAUST

Nii nagu igal inimesel on teatud ainuomane tunnusjoonte komplekt, läbi mille on võimalik teda määratleda ja kirjeldada, on ka igal veebilehel oma ainulaadne identiteet. See väljendub kõiges, mida veebileheküljel kuvatakse, mida omatakse ja mida väärtustatakse.

1.1 Google missioon

Google peab enda missiooniks koguda ja korrasta kogu maailma informatsioon ning teha see kättesaadavaks ja kasulikuks. (Google'i kodulehekülg, <http://www.google.com>, oktoober, 2009). Kaasaegsetele veebilehtedele on iseloomulikud nii rahvusvahelistumine kui mitmekeelsus. Veebilehed peavad tegelema erinevate huvigruppide ja kultuuridega. Ühiskonna infotehnoloogilise arenguga suureneb ka huvigruppide teadlikkus ja oskus enam nõudeid esitada: soovitakse kõrgemaid standardeid. Tekib ka nõudlus multifunktsionaalsete veebilahenduste järele.

1.1 Google (SOAP) Search API

SOAP (*Simple Object Access Protocol*) on minimaalne komplekt kokkuleppeid programmide käivitamiseks XML'i (*Extensible Markup Language*) abil üle HTTP (*HyperText Transfer Protocol*). SOAP-protokolli kasutades saab ühe operatsiooni süsteemi (näiteks Windows 2000) all töötav programm suhelda mitte ainult teistes arvutites sama opsüsteemi all töötavate programmidega, vaid ka teiste opsüsteemide (näiteks Linux) all töötavate programmidega (Vallaste 2009).

Google (SOAP) Search API (*Application Programming Interface*), mida on Google AJAX search API eellane, loodi arendajatele ja uurijatele, et nad saaksid kasutada Google otsingut oma programmides. Arendajad said kasutada kogu Google andmebaasi, et leida õige vastus 5

miljonite veebilehtede seest, mis olid struktureeritud. Nad said kasutada Google vahemälu ja kontrollida õigekirja. Ent toodet kasutada, oli vaja hankida toote jaoks võti, mis on enamikele Google toodetele vajalik.

1.2 Mida tähendab AJAX

AJAX (*Asynchronous JavaScript And XML*), sellega tähistatakse interaktiivsete veebirakenduste loomise meetodit, kus toimub jooksev kulissidetagune andmevahetus brauseri ja veebiserveri vahel, nii et kasutaja iga liigutuse peale pole vaja kogu veebilehte uuesti alla laadida. Märgistuskeelena kasutatakse XHTML'i (*Extensible Hypertext Markup Language*) ja veebilehe väljanägemist kirjeldatakse kaskaadlaadistike (*CSS- Cascading Style Sheets*) abil. Interaktiivsuse tagamiseks kasutatakse kliendipoolset skripti (peamiselt ECMAScript'i realiseerimine JavaScript või JScript) ja dokumendiobjektide mudelit. Asünkroonseks andmevahetuseks serveriga kasutatakse objekti XMLHttpRequest, vahel ka objekti IFrame (Vallaste 2009).v

1.3 Google AJAX search API

Detsembris 2006 lõpetas Google võtmete jagamise, siiski võtme olemasolul saab seda kasutada mõõndustega, teades, et rohkem uuendusi neile juurde ei tehta (Stocky 2009).

Selle asemel sai täieliku toetuse enne testimisel olnud Google AJAX search API rakendused. Kasutusele tulnud tooted olid mõeldud ainult veebilehitseja põhiseks JavaScript kasutamiseks, et kuvada otsingu tulemusi.

Mõnda aega elutses veel SOAP, kuna paljud ei saanud ega tahtnud kasutada JavaScript põhilist otsingut. Nemad jäid truuks SOAP Search API-le.

Kogu asi hakkas muutuma aprillis 2008, kui Google lisas otsingule RESTful täienduse, et toetada flashi ja mitte teisi JavaScripti keskkondi. Hetkel on need täiendused arendusjärgus ja pea iga päev tuleb uuendusi. Leitakse turvaauke ja samas ka uusi lahendusi otsingu kiiremaks muutmiseks. Aina enam veebilehtede programmeerijaid kasutavad Google AJAX search API tooted, kuna nad tahavad konkurentsipüüki.

1.4 AJAX search API võtme kasulikkus

Google AJAX search API laseb kasutada enda otsingut oma veebilehel, samas olles üks vähestest toodetest Google'i poolt, mis ei vaja tegevuseks võtit. Võtme kasutamine on vabatahtlik. Võtme kasutamisel otsingus annab vea ilmnemisel Google meeskonna toetuse, kes püüavad veebilehe omaniku probleemi lahendada.

Võtit saab registreerida, selleks on sul vaja minna veebilehele <http://code.google.com/apis/ajaxsearch/signup.html>. Soovitan läbi lugeda Google AJAX search API kasutamistingimused ja nendega nõustuda. Seejärel tuleb sisestada oma veebilehekülje nimi, vaja on sisestada ainult domeenimini ja Google laseb seda kasutada kõikidel sinu alamlehekülgedel. Ise kasutasin näites <http://www.tlu.ee>. Saadud võti on kõigile sama lehekülje kasutajale sama. Juhul kui oled enne võtme registreerinud, siis on see tegevus väga lihtne ja võtab aega mõned sekundid.

1.5 AJAX search API arhitektuur

Google AJAX search API arhitektuur on ehitatud andmete edastamisel läbi javascript koodi. Iga veebileht, mis on ühendatud internetiga võib andmeid küsida Google andmebaasist. JavaScripti kood võib olla lihtne nagu Search.execute(), kuid ka keerulisem. Vastused päringule saab XML formaadis või JSON formaadis. Kättesaadud andmete sõelumine võib toimuda käsitsi või automaatselt, kasutades selleks kasutajaliideseid, mis asuvad madalamates AJAX search API kihtides. Esposito (2009) sõnul on AJAX'is erinevate kihtide kasutamine uuenduslik, mida näitab ka just Google tegutsemine antud valdkonnas. AJAX search API arhitektuur pakub mitmed erinevaid funktsioone erinevatel kihtidel, millest teen ka kiire ülevaate.

Kogemustega programmeerijatele on mõeldud töötlemata kiht. Antud kihis tegutsemine nõuab häid oskusi JSON'is (JavaScript Object Notation) ja DOM'is (*Document Object*

Model) . Kuna vastavad teadmised saadakse üldiselt alles aastatega, siis siinkohal ei hakka antud kihti rohkem selgitama.

Keskmine kasutajaliidese kiht on mõeldud kasutajatele, kes soovivad AJAX search API põhilisi teenuseid kasutada. Programmeerija saab kasutada mõningaid lihtsaid funktsioone, nagu antud töös ka hiljem käsitlen.

Täiustatud kasutajaliidese kiht annab kasutajale kerge vaevaga enamus infot, mis tahetakse. Sellel tasemel on olemas kasutajaliidesed, mis on spetsialiseerunud konkreetsetele lahendustele, nagu *LocalSearch*, *VideoSearch* jne.

1.6 Täiustatud kasutajaliidese otsingud

Google on praeguseks hetkeks välja töötanud mitu erinevat kasutajaliidest täiustatud kihile.

Peamisteks kasutajaliidesteks on:

** LocalSearch*

Funktsioon Google MAP'le, millelt saad otsida kaardilt. Pean mainima, et otsingule „Tallinn University“ annab API vale koha. Otsingule „Tallinna Ülikool“ saab ilusasti kaardi pealt kõigi Narva mnt majade paiknemise.

** WebSearch*

See on kõige tavalisem Google otsing, sama võib leida www.google.com. Siinkohal ei hakka arvustama või kiitma Google otsingu üldist objektiivsust, sellest saab eraldi töö kirjutada.

** VideoSearch*

Saab otsida videoid, peamiselt Google enda video veebilehelt ja samuti ka <http://www.youtube.com>. Otsitakse läbi kogu Google videote andmebaas ja rõõmustav on, et esimese vastena otsingule „Tallinna Ülikool“ meie enda suhtekorralduse tehtud video „Ülikool on....“

** BlogSearch*

Otsib maailma juhtivatest veebipäevikutest vasteid sinu otsingu parameetritele. Tihti võivad otsingutulemused olla mõni päev vanad, kuna maailmas on palju veebipäevikuid, mida Google ei jõua indekseerida iga päev.

** NewsSearch*

Otsib uudiseid Google enda uudiste portaalist <http://news.google.com/>. Mis omakorda seirab uudiste portaale reaalajas.

** ImageSearch*

Kasutab piltide otsinguks täpselt sama süsteemi, mis <http://images.google.ee/>. Pildid kuvatakse ilusasti otsingu aknasse.

** BookSearch*

Selle otsingu tulemus suunab ümber kasutaja lehele <http://books.google.com/>. Jaanuar 2009 seisuga oli sinna skanneeritud üle 7 miljoni raamatu. Hetkel on seal saadaval enamasti inglise keelsed raamatud, samas on olemas enamus raamatute andmed. Näitena kasutasin A.Hitleri „Mein Kampf“. Üllatav oli, et seal oli isegi väike ülevaade raamatust, siiski ei olnud tegemist skanneeritud variandiga.

** PatentSearch*

Järjekordselt üks Ameerikast lähtuv Google rakendus. Nimelt otsib see Ameerika Patendiametis (United States Patent and Trademark Office) registreeritud tooteid.

1.7 Google AJAX search API Wizards

Google on teinud veebilehtede omanikele elu suhteliselt kergeks. Nimelt on nad valmistanud 7 viisardit enim kasutatavatele otsingutele. Kõik viisardid on üsna lihtsad ja mugavad ja neid saab enda nägemise järgi ka muuta.

1.8 Google AJAX Search API't internetis

Veebileht, mis asub aadressil <http://ajaxsearch.typepad.com/>. Seal lehel on olemas kolm Google AJAX search API otsingut: tavaline otsing, Google video otsing ja Google kaardi otsing. Seal õpetatakse kuidas teha veebipäeviku pidajatele erinevaid otsinguid.

Kaardile suunamise funktsioonist ka lähemalt.

Tuleb luua hüperlink, mis on JavaScriptina ja seal tuleb kasutada käsku `doMapSearch('otsingu tulemus');`. Kui sa otsid näiteks Tartut, siis sinu kood peaks välja nägema nii `Tartu`.

Töötab lihtsalt ja on võimalus teha väga palju huvitavat.

2. PRAKTILINE TEOSTUS

Antud peatükis kirjeldan läbi viidud praktilise teostuse aluseid, metoodikat ning väljundite ülevaadet. Näitena on toodud kolme otsingu kood. Üks on lihtne „Hello, World“ näide, teine on kaardi viisard ja viimane on isetehtud minimalistlik veebileht. Enne kui alustad, võta aega ja vaata Google API dokumentatsiooni lehel <http://code.google.com/apis/ajaxsearch/documentation/>. Seda tasub kõigil vaadata, sest tõenäoliselt saad sealt mõningaid huvitavaid ideid, kuidas kasutada Google teenuseid oma leheküljel.

2.1 "Hello, World"

Lihtsaim viis alustada õpingutega on kõigile tuttav "Hello, World". Ka siin saab samasuguseid paralleele tuua, aga siiski on see natuke keerulisem. Otsingu lähtekoodi (Lisa 1.) sain koos võtmega, natuke on seda modifitseeritud. Otsingu sõnaks kasutasin "Tallinn University".

2.1.1 Otsingu kood

Kogu kood on ehitatud *Hypertext Markup Language* (HTML) ja JavaScript koodis. Javascripti osa ehk otsingu mootor on üles ehitatud HEAD-i ja väljund harjumuspäraselt BODY-sse.

Alguses määrame ära dokumendi tüübi:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

Järgmisena tasub välja tuua, et kasutame xhtml'i:

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

Edasi liigume HEAD'i, kus nagu mainitud on üles ehitatud otsingumootori tuum. Kirjutame `<head>` ja `</head>`, nende vahele hakkame siis JavaScripti ehitama.

Enne veel tuleb määrata leheküljele tiitel ja HTML standard:

```
<meta http-equiv="content-type" content="text/html; charset=utf-8"/>
<title>Hello World - Google AJAX search API Näidis</title>
```

Nüüd alustame siis põhilisega, ehk otsingumootoriga:

```
<script src="http://www.google.com/jsapi" type="text/javascript"></script>
```

Lihtne tekst, aga väga oluline. Nimelt kaasasime just Google AJAX search API lähtekoodi.

Lähtekood olemas, lähme edasi ja lisame rea:

```
google.load('search', '1');
```

See selgitab Google serverile millist tema API-t kasutada tahame, hetkel on selleks „*search*“. Teine parameeter '1' on seal selle API versiooni number. Kusjuures iga Google API juures tuleb määrata ära ka tema versioon. Piisab teadmisest, et meie otsingu API on versioon 1, sest Google automaatselt pakub uusima 1 versiooni. Näiteks kui me paneksime versiooniks 1.1, siis Google annaks meile 1.1 kõige uuema versiooni. Vahel on sellest kasu, kui on teada, et piisab ainult mõnest teatud versioonist.

Teades mida me soovime, tahab server ka teada, mis otsinguid me serverist teha tahame.

Selleks käivitame funktsiooni:

```
function OnLoad() { }
```

Funktsiooni sees esimeseks loome otsingu kontrolli:

```
var searchControl = new google.search.SearchControl();
```

Edasi lisame just need otsingu tulemused, mida soovime kuvada. Vaata „Google otsingud“:

```
searchControl.addSearcher(new google.search.WebSearch());
```

Antud otsing teostab siis *WebSearch*'i. Selle asemel või samasugune uus rida luues, saab kasutada teisi otsinguid. Minu näites on kasutatud veel peale selle otsingu *VideoSearch*, *BlogSearch*, *NewsSearch*, *ImageSearch*.

Järgnev rida ütleb otsingule, et ta ennast kuvab ja määrab ära tema ID, kuna tegemist on erinevate otsingutega, siis on valitud ka ID'ks „searchcontrol“. Kui lehel on rohkem elemente tasub ka hoolikamalt mõelda, mida just teatud asja ID'ks määrata.

```
searchControl.draw(document.getElementById("searchcontrol"));
```

Lõpuni on veel natukene jäänud. Tuleb veel lisada otsingu parameeter ehk mida me otsida tahame:

```
searchControl.execute("");
```

Kui veebilehe omanik tahab, võib ta teha ka eelotsingu kasutaja jaoks. Ütleme, et tegemist on mingi mängu fänni-leheküljega, siis võib jutumärkide vahele kirjutada selle mängu nime ja otsing kuvab külastajale kohe info Googlest selle mängu kohta.

Aeg lõpetada ka script, lisame rea:

```
</script>
```

Olles JavaScriptiga valmis, lähme HEAD'st välja ja loome BODY:

```
<body>... </body>
```

Punktiiri asemele kirjutatakse veebilehtedel üldjuhul veebi kujundus. Seda teeme meiegi, lisades ennem ära märgistatud elemendi „searchcontrol“:

```
<div id="searchcontrol">Loading</div>
```

„Loading“ on seal näitena toodud, juhul kui interneti ühendus on aeglane või serveriga on mingi probleem. Põhimõtteliselt võib sinna teksti, pildi või mõne muu vajaliku rakenduse lisada.

Lõpetuseks peame sulgema HTML'i, selleks lisame viimase rea:

```
</html>
```

Nii ongi meie väike otsingumootor olemas (joonis 1).

Töötav lähtekood on nähtav (vt Lisa 1).

Tallinn University x
powered by Google™

▼ Web 

Tallinn University

36th EUCEN Conference “**University Lifelong Learning: Synergy between partners**”.


06-11-2008 ... **Tallinn University**, Narva mnt 25, 10120 **Tallinn**, ph 6409 101, ...

www.thu.ee

▶ Video (4) 

▶ Blog (4) 

▶ News (4) 

▼ Image 



Tallinn University

180 x 244

www.euroeducatio



Tango Tallinn

682 x 512

www.tango-

. . .

Joonis 1. „Hello, World“ otsingu „Tallinn University“ tulemuse kuvand

2.1.2 Harjutusi õpilasele

1. Loo otsingud kõigist Google otsingutest.
2. Otsi ainult pilte
3. „Loading“ teksti asemele lisa Google logo
4. Tee veebileht, kus kasutajad saavad ise valida, mis otsinguid nad kasutada tahavad

2.1.3 Kokkuvõte „Hello, World“ otsingule

Üldiselt on sellist otsingut väga lihtne oma veebilehele sobitada. Väikese vaeva juures saab juba arvestatava tööriista. Tuleks kindlasti lugeda veel natuke manuaale ja õpetusi ning selle otsinguga saab teha imesid.

2.2 Kaardi viisard

Järgmisena vaatleme, kuidas saada tööle viisard. (vt Lisa 2).

Viisardiks on valitud kaardi viisard. Selle võib leida aadressilt: <http://www.google.com/uds/solutions/wizards/mapsearch.html>.

2.2.1 Probleemid kaardi viisardiga

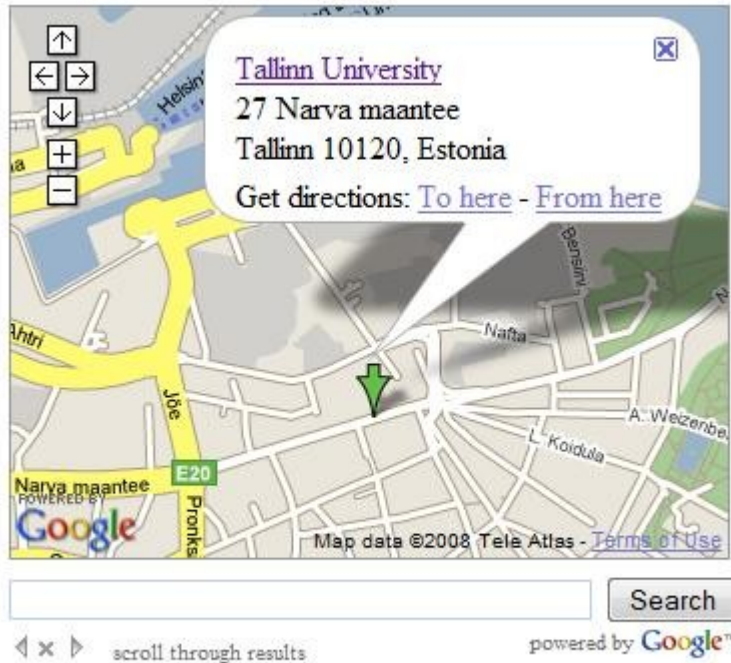
Alguses tundub kõik väga kena. Saab valida kaardi kõrgust ja laiust ning samuti kui kõrgelt soovetakse kaarti näha. "*Center location name*" tuleb panna koha nimi, mida soovid kaardil kuvada, kusjuures sinna võib panna suvalise nime. "*Center location link URL*" tekib esimene raskus, nimelt peab url täielikult välja kirjutama www.tlu.ee ei tööta, õige on <http://www.tlu.ee> "*Center location address*" tekib teine probleem. Kuna Google on enamjaolt USA põhine, siis ka siin tuleb kirjutada kaardilt otsitav aadress nagu USA-s. Et < kaardil Tallinna Ülikooli peamaja on vaja kirjutada sinna kasti " 27 Narva maantee, Tallinn" muud moodi soovitud tulemust ei suutnud leida.

Siis tekib järsku veel probleeme, nimelt kasutab see viisard peale Google AJAX search API ka Google maps API-t. Ja need mõlemad vajavad tasuta API võtit, mis on ühendatud Google Kontoga ja sinu lehekülje URL-ga ja kasutades neid API-si pead sa nõustuma mõlema kasutustingimustega.

Järjekordselt kasutan meie enda ülikooli veebilehte võtme saamiseks <http://www.tlu.ee>.

2.2.2 Otsingu tulemus

Saan otsingu JavaScripti ja väikese lõigu HTML koodi ehk siis väljundi otsingule. Seal on hästi ära selgitatud kuidas seda kasutada. Samuti jääb silma, et otsingule kui ka Mapsile on oma võti genereeritud.



Joonis 2. Kaardi viisardi tulemuse kuvand

2.2.3 Kokkuvõtte kaardi viisardile

Seadistamisega sain hakkama 2 minutiga (kogenenumad kasutajad teevad seda kindlasti kiiremini). Minule endale see raskusi ei tekitanud.

2.3 Veebileht kasutades Google AJAX search API't

Kogu veebileheküljel on ehitatud kasutades Google AJAX search API't ja näitab Google AJAX search API võimekust. Mainin, et leht näitab erinevaid meedia otsinguid Tallinna Ülikoolist ühel lehel. AJAX'i võimekus tuleb väga hästi just siin välja, kuna leht laeb alguses kõik talle antud otsingud.

Veebilehekülj koosneb kahest osast. Esimene osa on video otsingud <http://www.youtube.com> keskkonnast ja teine osa on <http://www.google.com> otsing.

2.3.1 Tallinna Ülikool meedias lehekülje kood

Siinkohal ei hakka kogu koodi järjest läbi käima, vaid tutvustanmuudatusi võrreldes eelmistega lühidalt.

Kood (vt Lisa 3) on jällegi ehitatud lihtsalt HEAD ja BODY sektsioonidega.

Seekord ei kasutanud otsingus võtit, et näidata ka ilma võtmeta kasutamist. Küll aga lisasin Google enda kujundus faili.

```
<link href="http://www.google.com/uds/css/gsearch.css" type="text/css" rel="stylesheet"/>
```

Samuti lisasin otsingud Twitterist ja Facebookist.

```
var twitter = new GwebSearch();
    twitter.setUserDefinedLabel("Twitter");
    twitter.setUserDefinedClassSuffix("twitter");
    twitter.setSiteRestriction("twitter.com");
```

Alguses defineerisin muutuja „twitter“, peale seda andsin talle vastava sildi klassi tunnuse ning lõpuks ka aadressi, kust andmed pärinevad.

Järgnevalt lisasin otsingu tulemuste arvu. Google lubab kasutada kõikidel otsingutel kahte surust *SMALL* ja *LARGE*. Esimene annab üldjuhul 4 tulemust ja teine 8. Ainult tavalisele veebi otsingule lubab Google ka kasutada *CSE* suurust, mis annab 10 vastust.

```
searchControl.setResultSetSize(GSearch.SMALL_RESULTSET);
```

Lisasin ka võimaluse otse leheküljel vaadata videoid. Tegemist on tavalise lehekülgedele lisatava koodiga, et näidata teiste lehekülgede videoid, ilma enda omalt lahkumata.

```
videoBar = new GSvideoBar(barContainer,
    GSvideoBar.PLAYER_ROOT_FLOATING, options);
```

Lõpuks tuleb lisada ka otsingu nupp ja vorm. Ei hakanud kirjutama otsingusse sisestusvälja otsingukriteeriume, kui soovi, siis soovitan selle lisada.

Selguse huvides ehitasin DIV'na BODY osa üles, aga juurde ei teinud eraldi css faili. Eelpool mainitud Google css fail annab vaid otsingute üldise kujunduse. Scudder mainib, et erinevate välimustega veebilehed võivad avaldada märkimisväärselt kasu. Seda nii kasutusmugavuselt, kui ka professionaalse või mängulise kujundusega.

Tallinna Ülikool meedias

Meeskoor ▾ Otsi



Tallinna Ülikool Search ×
powered by Google™

Web Veebikolumn **Twitter** FaceBook
News Raamatud Pildid

[Twitter / Tehnopol: Tehnopol ja Tallinna Ülikool](#)
Tehnopol ja Tallinna Ülikool sõlmisid koostööleppe. ... Tehnopol ja Tallinna Ülikool sõlmisid koostööleppe3:51 AM Aug 31st from web ...
[twitter.com](#)

[Tehnopol \(tehnopol\) on Twitter](#)
Tehnopol avab ukse Tallinna õpilastele
<http://bit.ly/3pkIry6>:51 AM Oct 19th ...
Tehnopol ja Tallinna Ülikool sõlmisid koostööleppe3:51 AM Aug 31st from web ...

Joonis 3. Tallinna Ülikool meedias

See on kõik sellest koodist. Seda koodi võib vabalt kasutada ja proovida.

2.3.2 Harjutusi õpilasele

1. Lisa juurde otsingusse leheküljed, millelt soovid otsida.
2. Loo CSS kujundus.
3. Luba otsida ka erinevaid videoid.

Kokkuvõte

Siinse töö eesmärgiks oli õppivale veebiehitajale uute ideede juurutamine nii, et just nende veebilehed oleksid külastajate seas populaarsed.

Tahtes muuta veebilehti efektiivsemaks, on vaja aru saada külastajate vajadustest kaasaja ühiskonnas. Vajadustele järgneb alati motiiv, stiimul, mis ergutab käitumist. Nii on veebilehtede omanikel võimalus jälgida hetke trende ja kuvada just seda oma veebilehel läbi Google AJAX search API, mis just külastajatele sobib (videod, kaardid, uudised). Saab olla suhteliselt kindel, et kui külastajat huvitavad uued naljakad videod näiteks <http://www.youtube.com>, siis lehekülg mis neid pakub on potentsiaalseks sihtmärgiks.

Külastajate pidev lehekülgede külastamine on juba üheks eelsoodumuseks, et ka hoiakud sarnaste veebilehtede suhtes oleksid positiivsed. Tänu tekkinud hoiakutele on võimalik ka kergema vaevaga kaardistada külastajate vajadused ja neid ka muuta.

Soovitan kõigil veebiprogrammeerijatel, kes tahavad oma veebilehel kasutada mingit otsingut mujalt veebilehekülgedelt kasutada just Google AJAX search API-t. Tema kasutamine on lihtne ja uusi võimalusi lisandub pea iga nädal.

Kasutatud kirjandus

H.Vallaste. (2009). e-Teatmik: IT ja sidetehnika seletav sõnaraamat. <http://www.vallaste.ee/>.

D.Esposito. (2009). Microsoft ASP.NET and AJAX: Architecting Web Applications. Microsoft Press. ISBN: 9780735626218.

J.Scudder. Custom Search with Custom Style: Peanut Butter and Jelly. <http://googleajaxsearchapi.blogspot.com/2009/08/custom-search-with-custom-style-peanut.html>. 26.10.2009.

E.Engel. *Google's Ajax Search API*. 27.10.2009. <http://www.stonetemple.com/articles/google-ajax-search-api.shtml>

T.Stocky. Introducing Labs for Google Code. 27.10.2009
<http://googlecode.blogspot.com/2009/03/introducing-labs-for-google-code.html>

Google AJAX Search API in TypePad. <http://ajaxsearch.typepad.com/>. 27.10.2009

LISAD

Lisa 1. Lähtekood "Hello, World" jaoks

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
2   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml">
4   <head>
5     <meta http-equiv="content-type" content="text/html; charset=utf-8"/>
6     <title>Hello World - Google AJAX Search API Näidis</title>
7     <script src="http://www.google.com/jsapi"
8       type="text/javascript"></script>
9     <script language="Javascript" type="text/javascript">
10    //<![CDATA[
11    google.load('search', '1');
12
13    function OnLoad() {
14      // Loob otsingu kontrolli
15      var searchControl = new google.search.SearchControl();
16
17      // Lisab otsingu tulemused
18
19      searchControl.addSearcher(new google.search.WebSearch());
20      searchControl.addSearcher(new google.search.VideoSearch());
21      searchControl.addSearcher(new google.search.BlogSearch());
22      searchControl.addSearcher(new google.search.NewsSearch());
23      searchControl.addSearcher(new google.search.ImageSearch());
24
25
26
27      // Käseb otsingul ennast kuvada ja kuhu ennast kinnitada
28
29      searchControl.draw(document.getElementById("searchcontrol"));
30
31      // Mida otsida
32      searchControl.execute("");
33    }
34    google.setOnLoadCallback(OnLoad);
35
36    //]]>
37  </script>
38 </head>
39 <body>
40   <div id="searchcontrol">Loading</div>
41 </body>
42 </html>
```

Lisa 2. Kaardi viisardiga saadud lähtekood

```
1 <!-- ++Begin Map Search Control Wizard Generated Code++ -->
2 <!--
3 // Created with a Google AJAX Search Wizard
4 // http://code.google.com/apis/ajaxsearch/wizards.html
5 -->
6
7 <!--
8 // The Following div element will end up holding the map search control.
9 // You can place this anywhere on your page
10 -->
11 <div id="mapsearch">
12   <span style="color:#676767;font-size:11px;margin:10px;padding:4px;">Loading...</span>
13 </div>
14 <!-- Maps Api, Ajax Search Api and Stylesheet
15 // Note: If you are already using the Maps API then do not include it again
16 //       If you are already using the AJAX Search API, then do not include it
17 //       or its stylesheet again
18 //
19 // The Key Embedded in the following script tags is designed to work with
20 // the following site:
21 // http://www.tlu.ee
22 -->
23 <script src="http://maps.google.com/maps?file=api&v=2&key=ABQIAAAATiIh3QRbRYw9dELv_
24   HB3MhRxedIj_hRAULd6lsTtNfCc75PKerQpYkvklquvGcfvDpVjd6dldLme9A"
25   type="text/javascript"></script>
26 <script src="http://www.google.com/uds/api?file=uds.js&v=1.0&source=uds-msw&key=ABQIAAAATiIh3QRbRYw9dELv
27   _HB3MhRxedIj_hRAULd6lsTtNfCc75PKerQpYkvklquvGcfvDpVjd6dldLme9A"
28   type="text/javascript"></script>
29 <style type="text/css">
30   @import url("http://www.google.com/uds/css/gsearch.css");
31 </style>
32
33 <!-- Map Search Control and Stylesheet -->
34 <script type="text/javascript">
35   window._uds_msw_dontorepair = true;
36 </script>
37 <script src="http://www.google.com/uds/solutions/mapsearch/gsmapsearch.js?mode=new"
38   type="text/javascript"></script>
39 <style type="text/css">
40   @import url("http://www.google.com/uds/solutions/mapsearch/gsmapsearch.css");
41 </style>
42
43 <style type="text/css">
44   .gmsc-mapDiv {
45     height : 275px;
```

```

46     }
47
48     .gsmc-idleMapDiv {
49         height : 275px;
50     }
51
52     #mapsearch {
53         width : 365px;
54         margin: 10px;
55         padding: 4px;
56     }
57 </style>
58 <script type="text/javascript">
59     function LoadMapSearchControl() {
60
61         var options = {
62             zoomControl : GSmapSearchControl.ZOOM_CONTROL_ENABLE_ALL,
63             title : "Tallinn University",
64             url : "http://www.tlu.ee",
65             idleMapZoom : GSmapSearchControl.ACTIVE_MAP_ZOOM,
66             activeMapZoom : GSmapSearchControl.ACTIVE_MAP_ZOOM
67         }
68
69         new GSmapSearchControl(
70             document.getElementById("mapsearch"),
71             "27 Narva maantee, Tallinn",
72             options
73         );
74
75     }
76     // arrange for this function to be called during body.onload
77     // event processing
78     GSearch.setOnLoadCallback(LoadMapSearchControl);
79 </script>
80 <!-- ++End Map Search Control Wizard Generated Code++ -->
81

```

Lisa 3. Veebileht „Tallinna Ülikool meedias“

```
1 <html>
2 <head>
3 <META HTTP-EQUIV="Tallinna Ülikool meedias" CONTENT="no_cache">
4
5 <meta name="keywords" content="Google AJAX search API">
6 <title>Tallinna Ülikool Meedias</title>
7
8 <link href="http://www.google.com/uds/css/gsearch.css" type="text/css" rel="stylesheet"/>
9 <script src="http://www.google.com/uds/api?file=uds.js&v=1.0&source=uds-vbw" type="text/javascript">
10 </script>
11   <script language="Javascript" type="text/javascript">
12
13     function OnLoad() {
14
15         var searchControl = new GSearchControl();
16
17         var uudised = new GlogSearch();
18         uudised.setUserDefinedLabel("Veebikolumn");
19
20         var raamatud = new GbookSearch();
21         raamatud.setUserDefinedLabel("Raamatud");
22
23         var pildid = new GimageSearch();
24         pildid.setUserDefinedLabel("Pildid");
25
26         var twitter = new GwebSearch();
27         twitter.setUserDefinedLabel("Twitter");
28         twitter.setUserDefinedClassSuffix("twitter");
29         twitter.setSiteRestriction("twitter.com");
30
31         var facebook = new GwebSearch();
32         facebook.setUserDefinedLabel("FaceBook");
33         facebook.setUserDefinedClassSuffix("facebook");
34         facebook.setSiteRestriction("facebook.com");
35
36
37         searchControl.setResultSetSize(GSearch.LARGE_RESULTSET);
38         searchControl.addSearcher(new GwebSearch());
39         searchControl.addSearcher(uudised);
40         searchControl.addSearcher(twitter);
41         searchControl.addSearcher(facebook);
42         searchControl.addSearcher(new GnewsSearch());
43         searchControl.addSearcher(raamatud);
44         searchControl.addSearcher(pildid);
```

```

44     searchControl.addSearcher(pildid);
45
46
47     var drawOptions = new GdrawOptions();
48     drawOptions.setDrawMode(GSearchControl.DRAW_MODE_TABBED);
49     searchControl.draw(document.getElementById("search_control"), drawOptions);
50
51
52     searchControl.execute("Tallinna Ülikool");
53
54     var videoBar;
55     var barContainer = document.getElementById("videoBar");
56     var playerContainer = document.getElementById("videoPlayer");
57
58     var options = {
59         horizontal : true,
60         largeResultSet : true
61     }
62
63     videoBar = new GSvideoBar(barContainer, GSvideoBar.PLAYER_ROOT_FLOATING, options);
64     videoBar.execute("Tallinna Ülikool");
65 }
66
67 </script>
68
69 <script src="http://www.google.com/uds/solutions/vidabar/gsvideobar.js" type="text/javascript">
70 </script>
71 <link href="http://www.google.com/uds/solutions/vidabar/gsvideobar.css" rel="stylesheet" type="text/css"/>
72
73
74 </thead>
75 <body onload="OnLoad()" >
76 <table border=0 cellpadding=0 cellspacing=0 height=250 width=100% margin-top=0>
77 <tr align=center>
78 <td height="250">
79 <table border=0 cellpadding=0 cellspacing=10 height="114" width="695" >
80 <tr>
81
82 <td colspan=2 height="6" width="675">
83
84 <h2>Tallinna Ülikool meedias</h2>
85
86 <form action="/" method="get">
87 <select onchange=submit() size="1" name="a">
88 <option value="Tallinna Ülikooli Meeskoor">Meeskoor</option>
89 <option value="Summer School at Tallinn">Suvekool</option>
90 <option value="Tallinna Ülikooli Kammerkoor">Kammerkoor</option>
91
92 </select>
93 <input type="submit" value="Otsi" >
94 </form>
95 <div id="videoBar"></div>
96 <div id="branding"></div>
97
98 <div id="search_control" align="middle" >Palun oota</div>
99
100 </td>
101 </tr>
102 </table>
103 </td>
104 </tr>
105 </table>
106 </body>
107 </html>

```